Method and Apparatus for the User-defined Loading and Running of Applications by Means of a Token

Field of the Invention

The present invention relates to a method and apparatus for the user-defined loading and running of applications by means of a token and in particular a chip-card.

Background of the Invention

Secure access to computers, i.e. the secure identification of the user, is the basis for virtually all the security provisions made in operating systems. Today, access to a system is safeguarded by means of password protection. To safeguard access to a computer system by using a chip card, there are extensions which need to be made to the architecture of the computer system. Each user whose identity code is to be protected by a chip card requires his own chip card on which the functions required (e.g. the encryption algorithm) and the relevant data are entered. A special device is needed to communicate with the chip card. This device, which is called a PINpad, comprises a reader unit, a keypad and a display. It is normally available to the user as an extra device additional to the keyboard and main display screen. The way in which access

protection by chip card is achieved is that an additional attribute can be specified for each identity code, namely whether access is now only to be possible by chip card and which users have access to the identity code in question. Within a computer there can be both identity codes which, as in the past, are safeguarded only by a password and identity codes which are safeguarded by a chip card as well.

Access to an identity code protected by a chip card is only permitted if the following conditions are met at the user interface:

- the user has successfully performed the logging-on procedure
- the user enters the correct PIN for the chip card
- the user is in possession of a chip card which matches the identity code in question.

The user logs on by entering a log-on string at the terminal. The computer is thus in a position to decide whether the identity code concerned is protected by a chip card or not. If it is, the user is asked to insert his chip card and enter the PIN via the PIN pad. The verification procedure then takes place.

Today there is a restriction on the widespread use of chip cards in that the systems used lay down rigid rules as to which applications can be run in what form with which chip cards. Only if the matching counterpart to the application is installed on the

chip card, which could even happen by chance, can the client use his chip card with the system concerned. If this is not the case, the application is unable to communicate with the chip card.

However, from the user's point of view, it is precisely when he wishes to vary the systems he uses that it would be desirable for any given system and its applications to orientate itself automatically to the particular user and his chip card rather than the other way around. It would then be perfectly possible to carry about with one a personal system with a customised configuration consisting of a variety of individual applications.

In the case of chip cards, there are standardised identifying mechanisms for making correlations between applications on the card-reading station (off-card applications) and their counterparts on a chip card (on-card applications). These are laid down in standards EMV 96 and ISO 7816. However, the idea underlying all such mechanisms is that the presentation and operation of the overall application which is shown to the client will always be determined by the off-card application. The chip card simply provides data, such as account number, name, address, etc. for one or more different applications. Hence, an application on an automatic account-keeping machine would be presented in the same form to all its authorised users. The option of varying the presentation, such as by varying the language in which the directions to the user are shown for example, would have to be explicitly programmed into the off-card application in a fixed

form. If nothing else, customers' preferences of this kind could be stored in permanent form as a notation in the bank's customer database or in a separate field on the card. The first however would not be possible if the customer belonged to some other bank and the second is a proprietary solution which could only be standardised for a few frequently chosen options (such as language) and for specific applications or sectors of commerce.

The object of the present invention is therefore to provide a method and apparatus which make the user of applications which are activated via a token independent of the local functionality of the input system.

Summary of the Invention

These and other objects of the invention are realized by a method and apparatus for the user-defining configuring and starting of an application or software components to form an application by means of a token and in particular a chip card. Via the service identifiers stored in his chip card which represent applications or software components to form applications, the chip card user can install the desired application. The card agent accepts the request for an application, checks the register to see if the application is present and if it is not, makes a connection to communicate with the server in order to download the application to the user system.

Brief Description of the Drawings

The present invention will now be described by reference to preferred embodiments thereof and by reference to the figures, in which:

Fig. 1 shows the apparatus according to the invention; and

Fig.2 shows the apparatus according to the invention in a JAVA environment.

Detailed Description of the Invention

Fig.1 shows the inventive basic principle of the present invention. Unique identifiers (service identifiers) are stored on a chip card in a non-volatile memory. The service identifiers identify runnable applications, or runnable software components which can be combined to form an application. The storage of the service identifiers on the chip card can take place the first time a given application is called up and it will preferably be done automatically. In another embodiment, the chip card may be supplied with certain service identifiers already provided. The service identifiers may for example comprise numbers or strings of characters, such as a GUID (global unique ID) or a URL, which are

stored on the chip card in a file. As well as this, the service identifiers may also include additional data relating to the application and the card-holder's preferences. The applications or software components form the link between the chip card and the user (the user interface of the application at the card-reader end) or other systems (databases, networks, etc.). They are preferably stored on the user system or on the system that the user regularly works with. In Fig.1, service identifiers ~~1 – 3~~ 101-103 are stored on the chip card. Software components ~~1-2~~ 111-113 103 are stored on the user system 120 with which the chip card directly communicates. Also installed on this system are a card agent 105 and a register. 106

If the customer inserts the chip card 100 in a card reader belonging to the user system, the card agent 108, 107 gives an APDU command to the chip card for it to communicate the application request and the minimum requirements of the application. The card agent has the appropriate drivers for the various card technologies (e.g. Mondex, (Watshaw) JavaCard, ISO) and having determined the card technology it loads the requisite driver. By referring to the communication (service identifier) from the chip card, the card agent checks whether the application or software component requested is available on the local user system and whether the card reader can meet the requirements of the application. The card agent retrieves the appropriate application or software component via the register. The register, which is implemented in the form of a file, table or database, manages all the available applications or software

components which are offered on this specific user system/card reader. Where a certain software component is not available locally on the user system/card reader, there are two possibilities.

First, the desired software component can be loaded from a server specified on the chip card. When this is the case, the service identifier will include the exact address of the server on which the desired software component is stored. If the card agent finds that the desired software component is not contained on the register, it makes a connection via the network to the relevant server and downloads the desired software component to the user system/card reader.

Alternatively, by referring to information relating to the software component which is stored on the chip card, the user system/card reader can decide whether it is able to offer similar or equivalent software components which meet the same specification.

In a further embodiment, the chip card may contain not only the service identifiers for the particular applications but also service identifiers for the card agent and the register. If the chip card user wishes to log on to some outside system on which neither the card agent nor the register is stored, he must first download the two modules which are stored on his user system to the system he currently wishes to work with. For this it is necessary for the service identifiers to contain the address information for these modules (e.g. URL's = Universal Resource Locations). The

modules having been downloaded, the desired application is then run in the same way as was described in detail above.

In the case of an account-keeping application for example, various software components may be available which all use the same interface conforming to a predefined specification and which, where required, hold certain certificates. With the appropriate service identifier the chip card user can select a given account-keeping component. The suppliers of such software components may be the company bank or an independent software house. It is also possible for individually programmable software components to be offered (the swatch concept). Apart from different and personalised presentations at the user interface (e.g. language, positioning or clarity of layout), different software components can also implement slightly different applicational logic. For example, account-keeping tool A may sort transfers only by date whereas software component B may also allow them to be displayed arranged by amount. In another embodiment, a customer may combine various applications, such as the management of his account by his bank, his mobile telephone account with a telephone company and the booking of flights by a travel agent, into one overall application.

An off-card section on the chip card comprises service identifiers for the desired off-card applications (software components) and data conforming to the accustomed specifications such as SKA, GSM and IATA. The individual software components are combined in background without the user having to do anything. The

chip card can call up the appropriate software components by using predefined interfaces. This transfer of responsibility is particularly helpful because the chip card itself knows which applications are stored on it and are immediately accessible to its holder.

Constructing the off-card application oneself in customised form would be considerably more complicated because this part of the application is only indirectly accessible to the user. Settings and variants would have to be individually programmed and settings would have to be stored separately in the background systems.

Apart from the greater flexibility obtained and the focus on the requirements of the customer, it is also possible in this way to implement plug-and-play mechanisms which enable any desired cards to be used at any desired reader stations.

Fig. 2 shows an actual implementation of the present invention in a JAVA environment. In this case there are three software components combined into one application. The first software component is a user interface, which has been produced by means of JAVA AWT for example. The second component is formed by mechanisms for accessing a database using VisualAge Data Access Beans and the third component is used for accessing a server by means of Java Enterprise Beans or even servlets. Mechanisms for the worldwide allocation of identifiers are known (e.g. GUID, URL). The service identifiers for the present software components (services 1 - 3) are situated on the chip card.

The idea on which the present invention is based is therefore to combine or configure the desired applications by means of a token and in particular a chip card. Via the service identifiers stored in his chip card which represent various applications or settings for applications, the chip card user can select the desired applications or settings and install them. The card agent accepts the request for an application, checks the register to see if the application is present and if it is not, makes a connection to communicate with the server in order to download the application to the user system.